

# Pengenalan Java

---

## OBJEKTIF :

1. Mahasiswa mampu belajar Bahasa Pemrograman Java seperti mengetahui Bahasa Pemrograman Java, langkah Instalasi dan pengaturan JDK, mengetahui *Integrated Development Environment*, menulis Program Java, menganalisa kerangka Program Java, comment serta error pada Bahasa Pemrograman Java.
  2. Mahasiswa mampu menginstal JDK 17 dan Apache Netbeans 13.0.
  3. Mahasiswa mampu menggunakan *Software IDE* Netbeans untuk menulis Program Java.
- 

## Pengertian dan Sejarah Java

---

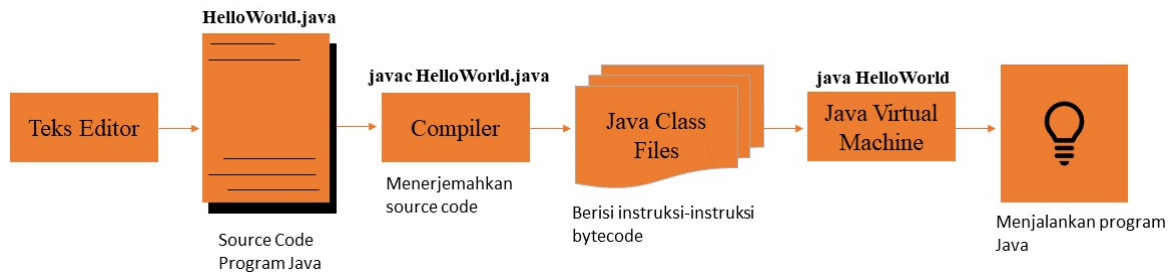
Bahasa Java merupakan bahasa pemrograman berorientasi objek yang dapat dijalankan di semua perangkat seperti komputer serta telepon genggam. Java dibuat oleh James Gosling saat ia masih bekerja di Sun Microsystem, yang sekarang perusahaan Sun Microsystem sudah diakuisisi oleh Oracle. Java pun dirilis ke publik pada tahun 1995.



Bahasa Java sendiri terkenal dengan jargonnya yaitu "*Write once, run anywhere*". Yang berarti satu source code Java dapat dijalankan di semua perangkat yang sudah terinstall JVM (Java Virtual Machine). Hal tersebut dapat terjadi karena setelah *source code* Java tercompile akan berubah menjadi *bytecode* yang dapat berjalan independen di atas JVM.

## Compiler dan Java Virtual Machine

Untuk membuat program Java kita bisa mengetikannya pada code editor ataupun IDE dan menyimpannya menjadi sebuah file yang berekstensi .java. Kode yang sudah kita buat tersebut disebut sebagai *Source Code*.



Setelah kita menyimpan *source code* tersebut selanjutnya kita harus menjalankan *compiler* Java. *Compiler* merupakan sebuah program yang menerjemahkan *source code* menjadi instruksi-instruksi yang dapat dijalankan komputer atau bisa disebut juga sebagai *bytecode*. Dalam proses penerjemahan tersebut, *compiler* dapat memberitahukan error-error (kesalahan-kesalahan) yang terdapat dalam *source code*. Error-error tersebut harus diperbaiki agar proses penerjemahan dapat berjalan sukses.

Untuk menjalankan *source code* yang sudah diterjemahkan menjadi *bytecode*, kita harus menggunakan JVM atau Java Virtual Machine. JVM sendiri merupakan sebuah program yang mensimulasikan sebuah komputer yang bahasa mesinnya merupakan *bytecode*.

## Instalasi dan Pengaturan JDK

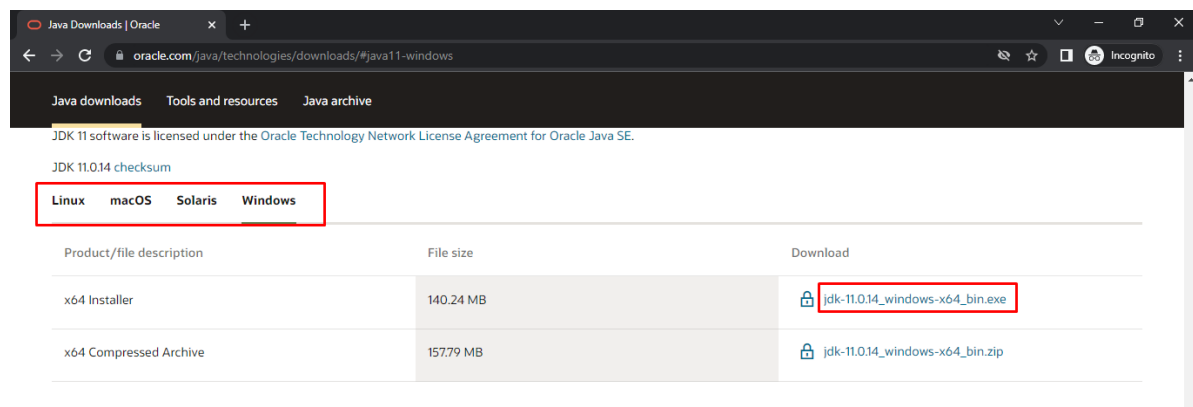
Untuk melakukan proses pengembangan program Java, kita memerlukan *compiler* Java dan JVM. Keduanya tersedia dalam satu paket yang disebut dengan JDK (Java Development Kit).

### Proses Instalasi JDK

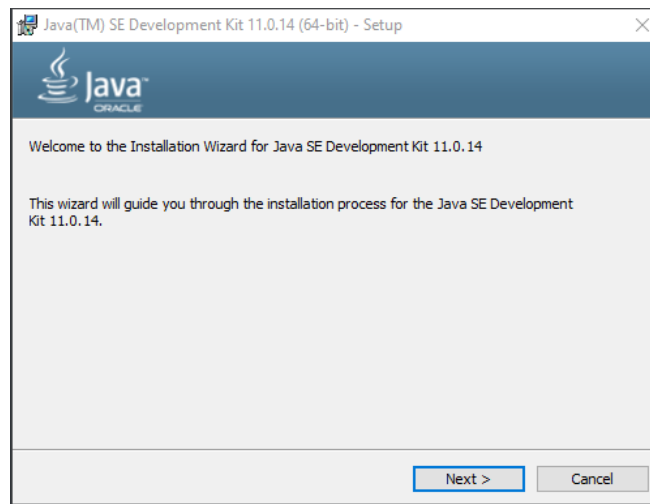
1. Lakukan pengunduhan *installer* JDK pada website resmi Oracle atau melalui link berikut.

<https://www.oracle.com/java/technologies/downloads/#java11>

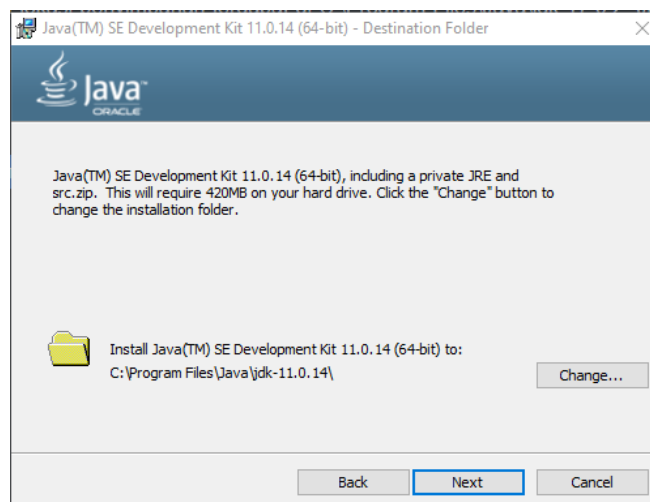
Lalu pilih sistem operasi yang Anda gunakan. Sebagai contoh, dilakukan pengunduhan *installer* Java SE untuk sistem operasi Windows dengan memilih salah satu link pada kolom *download* lalu proses *download* akan dimulai.



2. Setelah file berhasil *download*, buka file JDK lalu akan muncul jendela *Setup*. Untuk memulai proses instalasi klik *Next* pada jendela *Setup*.



Setelah itu Anda diminta untuk menentukan lokasi penyimpanan folder instalasi di direktori perangkat Anda. Secara default lokasi penyimpanan terdapat di C:\Program Files\Java\jdk-17.0.2. Jika ingin merubah lokasi penyimpanan folder instalasinya Anda bisa memilih tombol *Change*, jika tidak ingin dirubah maka Anda dapat memilih tombol *Next*.



Lalu tunggu proses instalasi selesai.

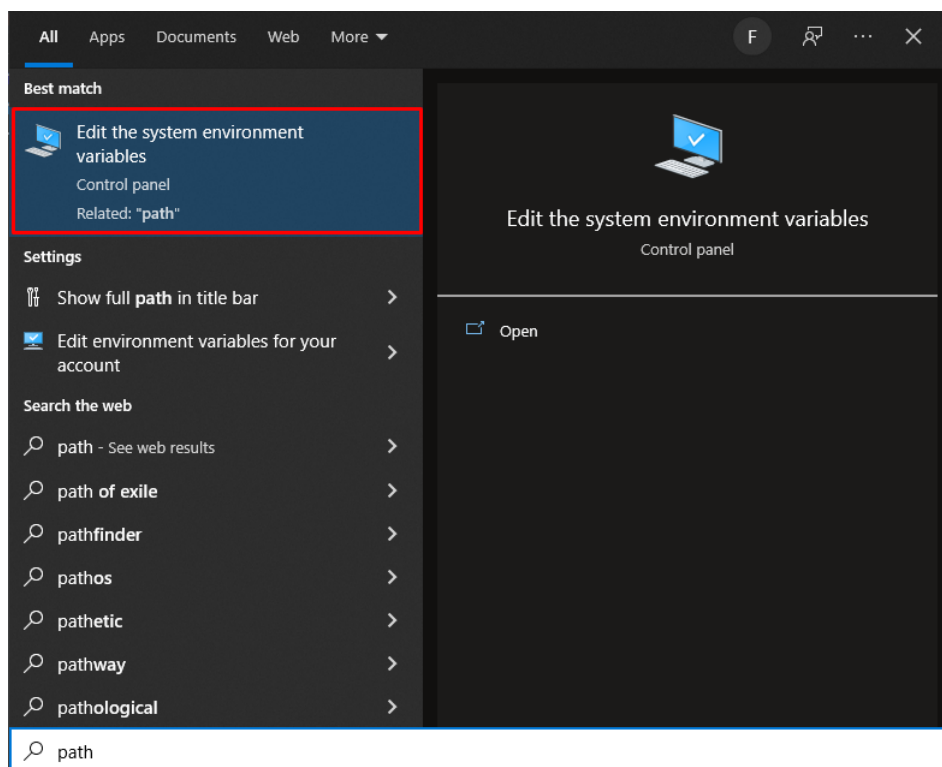
Tahapan proses instalasi berhasil, jika *instaler* menampilkan tampilan seperti dibawah ini. Klik Close untuk menutup *instaler*.



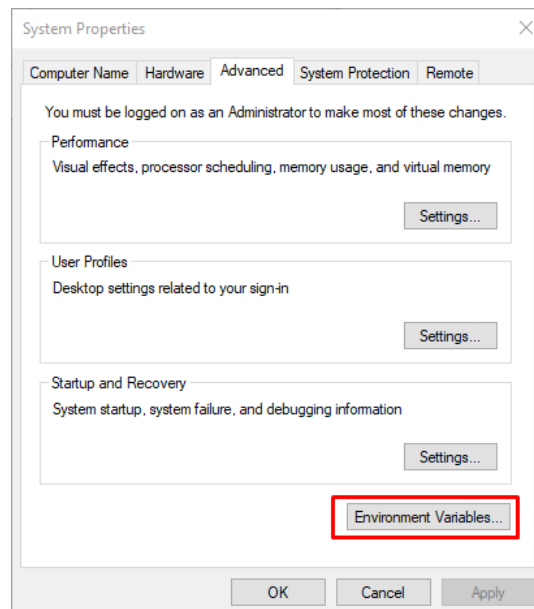
## Mengatur Path Variable Environment

Setelah proses instalasi JDK selesai, langkah selanjutnya adalah mengatur *path* JDK pada *variable environment* Anda.

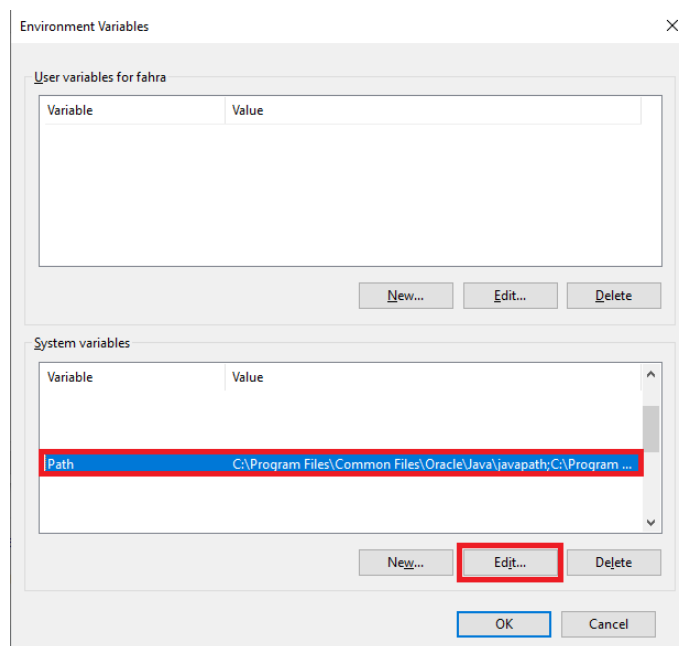
1. Buka *Search Bar* Anda lalu ketikkan *path* maka akan muncul best matches seperti berikut. Lalu klik pada bagian *Edit the system environment variables*.



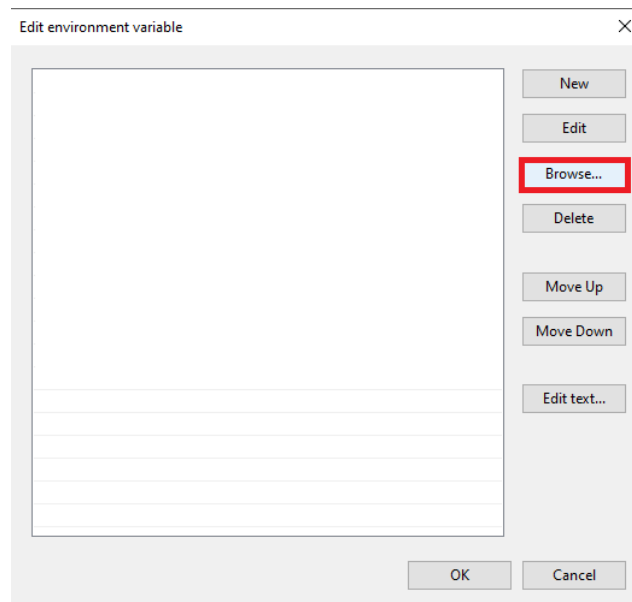
2. Setelah itu, jendela *System Properties* muncul lalu anda bisa memilih bagian *Environment Variables*.



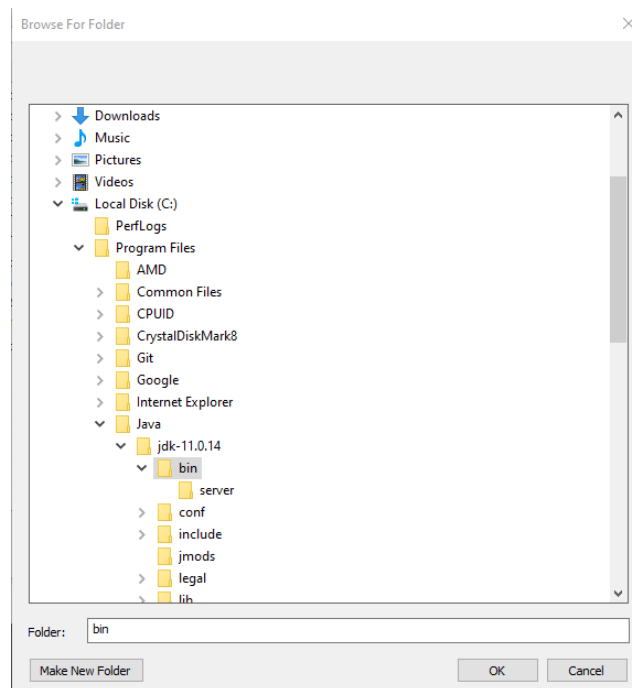
3. Selanjutnya jendela *Environment Variables* terbuka, pilih *Path* yang terdapat pada bagian *System variables* lalu pilih *Edit*



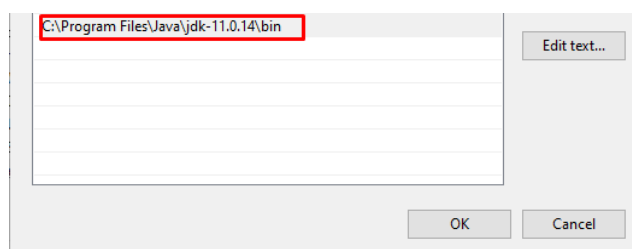
4. Setelah itu pilih *Browse* untuk menambahkan *path* dari lokasi penyimpanan folder JDK Anda.



5. Selanjutnya arahkan ke folder `bin` pada lokasi penyimpanan folder JDK anda dan klik OK.



6. Setelah itu path dari lokasi penyimpanan folder JDK anda pun akan muncul didalam jendela *Edit environment variables*.



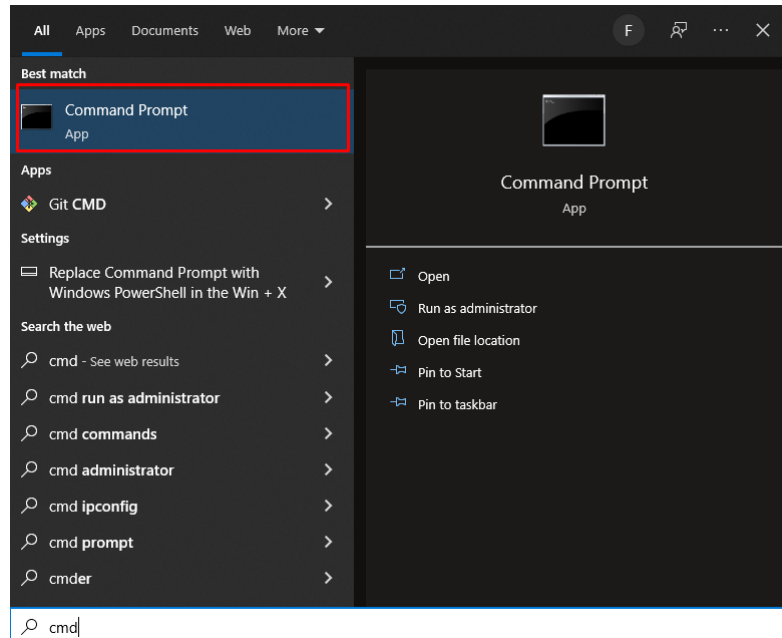
Selanjutnya Anda bisa memilih OK.

Dengan ini Anda telah berhasil mengonfigurasi path JDK pada *Environment Variable*.

## Cara mengetahui JDK telah berhasil di instal

Setelah berhasil mengonfigurasi path JDK pada *Environment Variable*, selanjutnya Anda harus memastikan bahwa JDK telah berhasil terinstal di device Anda.

1. Langkah pertama buka *Command Prompt* dengan mencarinya dalam *Search Bar* perangkat Anda.



2. Selanjutnya, ketikkan perintah berikut pada *Command Prompt*

```
java --version
```

```
C:\Users\fahra>java --version
java 11.0.14 2022-01-18 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.14+8-LTS-263)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.14+8-LTS-263, mixed mode)
```

Jika Java berhasil terinstal, maka akan menampilkan tulisan seperti diatas.

## Instalasi IDE

IDE atau *Integrated Development Environment* merupakan software yang digunakan untuk membantu proses pembuatan sebuah program. Saat membuat sebuah program yang tidak terlalu rumit seperti *console application*, kita cukup menggunakan aplikasi teks editor untuk membuatnya, untuk membuat program yang lebih kompleks seperti program berbasis GUI kita perlu bantuan IDE untuk mempermudah proses pembuatan program kita.

Pada praktikum kali ini, IDE yang kita gunakan adalah NetBeans. Berikut merupakan panduan instalasi NetBeans.

1. Lakukan pengunduhan *installer* NetBeans pada link berikut.

```
https://netbeans.apache.org/download/nb120/nb120.html
```

2. Pilih sesuai sistem operasi yang anda gunakan. Sebagai contoh, dilakukan pengunduhan *installer* Java SE untuk sistem operasi Windows.

# Downloading Apache NetBeans 12.0

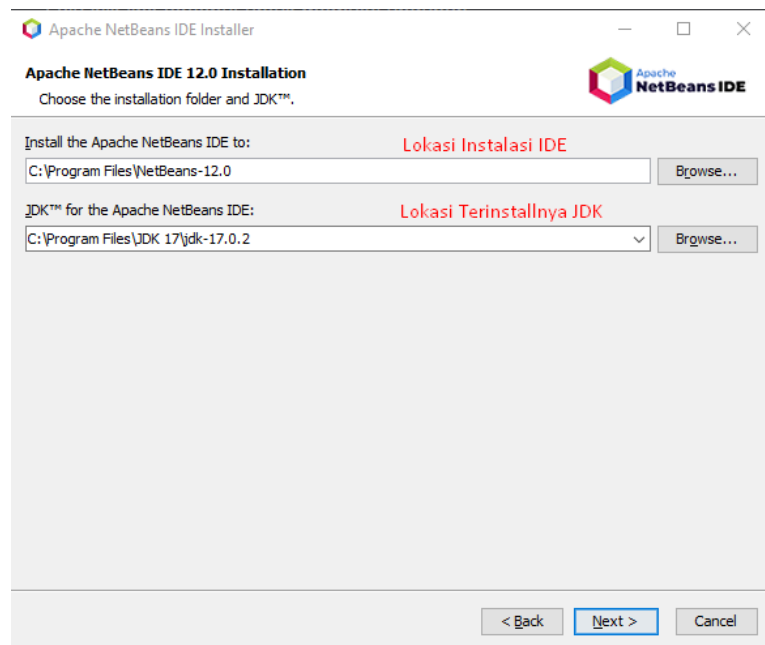
Apache NetBeans 12.0 was released on June 4, 2020. See [Apache NetBeans 12.0 Features](#) for a full list of features.

Apache NetBeans 12.0 is available for download from your closest Apache mirror.

- Binaries: [netbeans-12.0-bin.zip](#) (SHA-512, PGP ASC)
- Source: [netbeans-12.0-source.zip](#) (SHA-512, PGP ASC)
- Installers:
  - [Apache-NetBeans-12.0-bin-windows-x64.exe](#) (SHA-512, PGP ASC)
  - [Apache-NetBeans-12.0-bin-linux-x64.sh](#) (SHA-512, PGP ASC)
  - [Apache-NetBeans-12.0-bin-macosx.dmg](#) (SHA-512, PGP ASC)

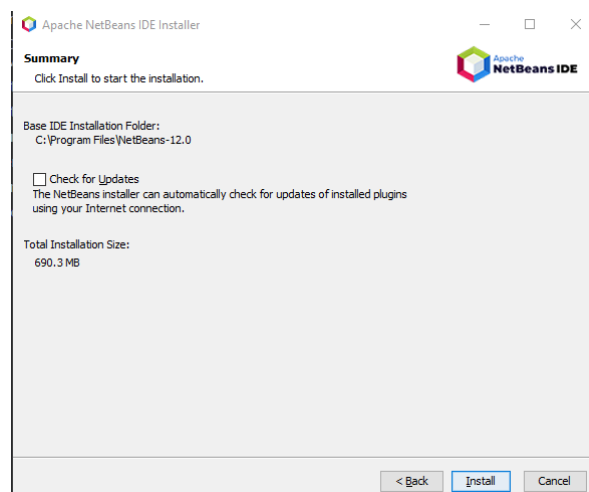
Lalu klik *link* tersebut untuk memulai unduhan

3. Setelah file berhasil diunduh, buka file *instaler* NetBeans lalu klik Next pada setiap windows sampai muncul tampilan seperti berikut. Selanjutnya sesuaikan lokasi instalasi IDE dan lokasi JDK anda.



Setelah lokasi sudah disesuaikan klik *next*.

4. Setelah itu hilangkan centang pada bagian *Check For Updates* lalu klik install.



Tunggu sampai proses instalasi selesai, lalu NetBeans sudah bisa Anda gunakan.



# Kerangka Program Java

---

Untuk membuat program Java pertama, Anda harus mengetahui seperti apa kerangka dasar dari program Java.

```
public class NamaClass
{
    public static void main(String[] args)
    {
        ...
    }
}
```

Untuk membuat program baru menggunakan Java, anda dapat mengganti `NamaClass` dengan nama yang Anda mau selanjutnya Anda bisa menuliskan *statement-statement* dalam body dari method `main`. Setelah itu simpan source code anda dan pastikan nama file source code anda sama dengan `NamaClass` Anda.

## Menulis Program Java Pertama

---

Setelah Anda memahami kerangka dasar dari program Java, selanjutnya Anda dapat membuat program Java pertama Anda. Bukan hanya menulis *source code* nya saja, tetapi Anda juga akan belajar cara untuk mengcompile source code tersebut lalu menjalankannya.

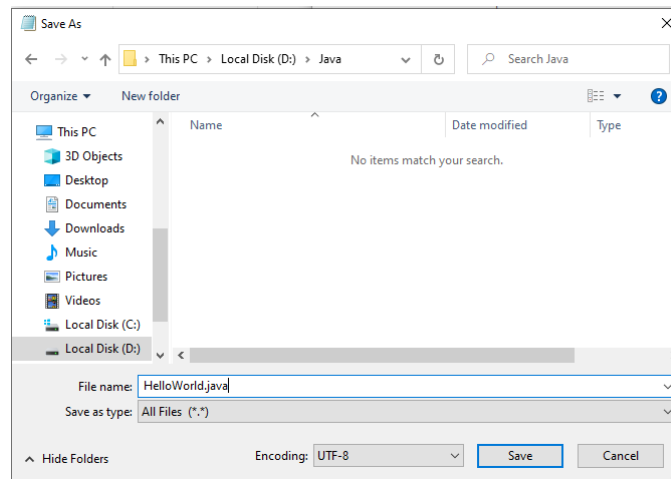
**Langkah 1.** Buka *teks editor* anda lalu ketikkan kode Java berikut.

**Program (HelloWorld.java)**

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello, world!");
    }
}
```

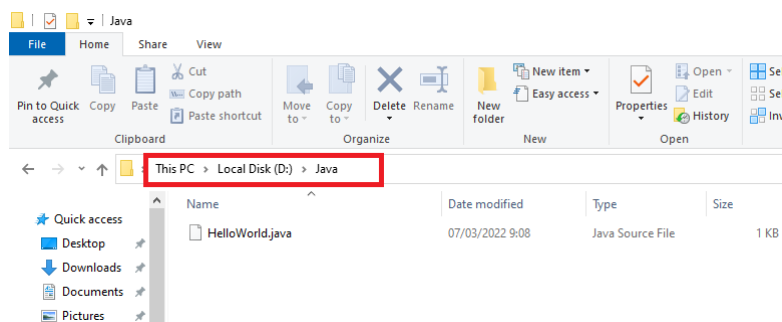
Untuk membuat teks terindentasi atau menjorok kedalam, Anda bisa mengklik `tab` pada *keyboard* Anda.

**Langkah 2.** Simpan *source code* tersebut dengan ekstensi `.java`.

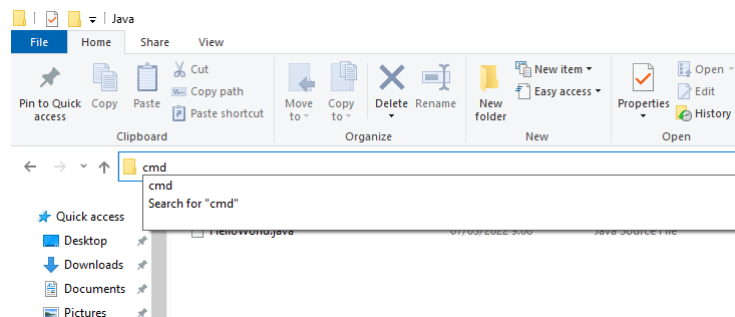


**Langkah 3.** Setelah *source code* tersimpan, selanjutnya kita akan mengcompile *source code* tersebut menjadi *bytecode*.

Pergi ke lokasi *source code* Anda.



**Langkah 4.** Ganti alamat folder Anda dengan cara mengklik daerah yang diberi kotak merah, lalu hapus alamat folder Anda dan ketikkan `cmd` setelah itu klik enter.



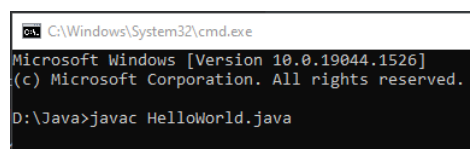
Maka jendela *Command Prompt* akan terbuka di lokasi penyimpanan *source code* Anda.

**Langkah 5.** Selanjutnya Anda harus mengetahui cara untuk mengcompile *source code* Anda. Ketikkan perintah berikut pada command prompt untuk mengcompile *source code* anda menjadi *bytecode*.

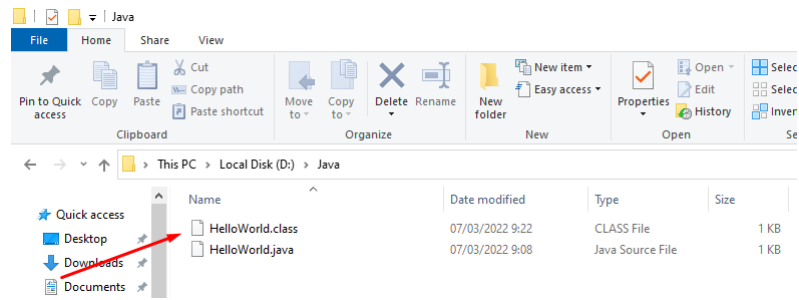
```
javac NamaFile.java
```

Mengcompile *source code* HelloWorld.java yang sudah dibuat.

```
javac HelloWorld.java
```



Setelah program berhasil di compile, maka akan terbuat file baru yang berekstensi .class. File tersebut merupakan *source code* yang sudah di terjemahkan menjadi *bytecode*.



## Langkah 6. Menjalankan *bytecode* menggunakan JVM.

Untuk menjalankan *bytecode* menggunakan JVM, anda bisa mengetikkan perintah berikut pada *Command Prompt*.

```
java NamaFile
```

Dapat dilihat bahwa saat kita ingin menjalankan *bytecode* menggunakan JVM, kita tidak perlu menulis ekstensi dari *bytecode* tersebut yaitu .class.

Menjalankan *bytecode* dari HelloWorld.

```
java HelloWorld
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

D:\Java>javac HelloWorld.java

D:\Java>java HelloWorld
Hello, World!
```

Setelah program dijalankan, maka akan muncul teks Hello,World!. Teks tersebut dapat muncul karena Anda telah mencetak teks tersebut dalam *source code* Anda menggunakan perintah .

```
System.out.println("Hello, world!");
```

## Comment

*Comment* atau Komentar merupakan bagian code Java yang tidak akan tereksekusi atau tercompile. *Comment* umumnya digunakan untuk memberikan catatan singkat pada suatu bagian kode. *Comment* berfungsi untuk pengingat bagi diri kita sendiri atau bagi programmer lain yang bekerja dengan code kita.

Terdapat dua cara penulisan *comment* yaitu, comment satu baris dan *comment* banyak baris.

Berikut merupakan contoh penulisan *comment* satu baris.

```
// Ini merupakan Single Line Comment
```

Selanjutnya comment multi baris, comment multi baris ditandai dengan memberikan simbol `/*` dan ditutup dengan simbol `*/`.

Berikut contoh penulisan comment banyak baris.

```
/* Ini  
Merupakan Multi  
Line Comment  
*/
```

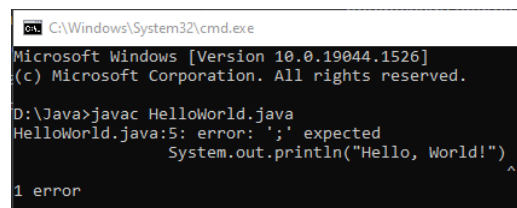
## Error

Tidak selamanya program yang Anda buat berjalan berhasil tanpa ada *error* satupun, karena dalam proses pengetikkan *source code* bisa saja terjadi *human error* yang membuat code yang sudah Anda tulis mengalami kesalahan. Jika terjadi kesalahan dalam pengetikkan code maka disebut **Syntax Error**, salah satu kesalahan *syntax error* ialah kesalahan eja atau lupa menulis `;` pada akhir statement. Jika terjadi *error* seperti *sytanx error*, maka program tidak akan tercompile melainkan compiler akan memberitahu Anda bahwa terdapat *error-error* yang Anda harus perbaiki terlebih dahulu sebelum mengcompile.

Sebagai contoh, misalkan pada program berikut, Anda lupa memberi `;` pada akhir *statement* `System.out.println`

```
public class HelloWorld  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hello, World!")  
    }  
}
```

Maka setelah Anda mencoba untuk mengcompile program tersebut, akan muncul pesan bahwa program tidak berhasil dicompile karena terdapat *syntax error* pada program.



```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 10.0.19044.1526]  
(c) Microsoft Corporation. All rights reserved.  
  
D:\Java>javac HelloWorld.java  
HelloWorld.java:5: error: ';' expected  
        System.out.println("Hello, World!")  
                                ^  
1 error
```

## REFERENSI

- [1] Horstmann, Cay S. 2012. *Big Java: Late Objects, 1st Edition*. United States of America: John Wiley & Sons, Inc.
- [2] Gaddis, Tony. 2016. *Starting Out with Java: From Control Structures through Objects (6th Edition)*. Boston: Pearson.